

WYDZIAŁ ELEKTRONIKI	
KARTA PRZEDMIOTU	
Nazwa w języku polskim:	Praktyczne aspekty wytwarzania oprogramowania
Nazwa w języku angielskim:	Practical Aspects of Software Production
Kierunek studiów:	Automatyka i robotyka, Elektronika, Informatyka, Telekomunikacja, Teleinformatyka
Stopień studiów i forma:	I stopień / II stopień, stacjonarne
Rodzaj przedmiotu:	wybieralny
Kod przedmiotu	ETEW16011
Grupa kursów:	TAK

	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium
Liczba godzin zajęć zorganizowanych w Uczelni (ZZU)	30		30		
Liczba godzin całkowitego nakładu pracy studenta (CNPS)	60		60		
Forma zaliczenia	Zaliczenie na ocenę		Zaliczenie na ocenę		
Dla grupy kursów zaznaczyć kurs końcowy (X)	X				
Liczba punktów ECTS	3				
w tym liczba punktów ECTS odpowiadająca zajęciom o charakterze praktycznym (P)	-		2		
w tym liczba punktów ECTS odpowiadająca zajęciom wymagającym bezpośredniego kontaktu (BK)	1		1		

WYMAGANIA WSTĘPNE W ZAKRESIE WIEDZY, UMIEJĘTNOŚCI I INNYCH KOMPETENCJI

1. Podstawowa umiejętność pracy w tekstowej konsoli Linux

CELE PRZEDMIOTU

- C1 Nabycie podstaw teoretycznych pracy w projekcie przy użyciu metodyk zwinnych.
- C2 Nabycie podstawowej wiedzy w zakresie zcentralizowanych systemów kontroli wersji na przykładzie Subversion.
- C3 Nabycie wiedzy z zakresu rozproszonych systemów kontroli wersji.
- C4 Nabycie podstawowej wiedzy w zakresie technik i narzędzi umożliwiających ciągłą integrację oprogramowania (Continuous Integration).
- C5 Poznanie podstawowych konstrukcji programistycznych w języku Python.
- C6 Zdobyć umiejętność praktycznego posługiwania się metodologią Scrum w zarządzaniu projektami informatycznymi.
- C7 Zdobyć umiejętność praktycznego posługiwania się systemami kontroli wersji w codziennym procesie tworzenia oprogramowania. Nabycie praktycznych umiejętności z zakresu tworzenia nowych repozytoriów, konfiguracji, zarządzania gałęziami.
- C9 Zdobyć umiejętność posługiwania się narzędziami zapewniającymi ciągłą integrację ze

szczególnym uwzględnieniem narzędzi Jenkins/Hudson.

C10 Nabycie podstawowych umiejętności programowania obiektowego w języku Python.

C11 Nabycie podstawowej umiejętności pracy w tekstowej konsoli Linux

C12 Nabycie podstawowych umiejętności konstrukcji programistycznych w języku Bash.

C13 Zdobycie praktycznych umiejętności rozwiązywania zadań programistycznych przy użyciu technologii języków skryptowych.

C14 Nabycie podstawowych informacji na temat architektury chmury obliczeniowej.

C15 Nabycie podstawowych umiejętności modelowania zależności pomiędzy plikami źródłowymi przy użyciu GNU Make.

PRZEDMIOTOWE EFEKTY KSZTAŁCENIA

Z zakresu wiedzy:

PEK_W01 Posiada wiedzę o cyklu życia projektu i metodach zarządzania kompleksowymi projektami informatycznymi.

PEK_W02 Posiada wiedzę z zakresu scentralizowanych i rozproszonych systemów kontroli wersji, ich charakterystyk i podstawowych różnic.

PEK_W03 Posiada wiedzę dotyczącą ciągłej integracji (CI), jej znaczenia w cyklu życia oprogramowania oraz wpływu CI na jakość wytwarzanego oprogramowania.

PEK_W04 Zna specyfikę tworzenia aplikacji obiektowych w języku Python.

PEK_W05 Posiada wiedzę dotyczącą podstaw poruszania się w Linuxie.

PEK_W06 Posiada wiedzę dotyczącą podstaw programowania w języku Bash

PEK_W07 Zna podstawowe pojęcia związane z cloud computingiem.

PEK_W08 Potrafi wskazać zależności pomiędzy plikami źródłowymi oraz opisać je używając składni GNU Make.

Z zakresu umiejętności:

PEK_U01 Potrafi użyć metodologii Scrum do prowadzenia projektu informatycznego.

PEK_U02 Potrafi wykorzystać podstawowe systemy kontroli wersji do zarządzania jakością i integralnością oprogramowania.

PEK_U03 Potrafi skonfigurować i wykorzystać narzędzia ciągłej integracji oprogramowania do kolejkwania i zrównoleglenia budowy oprogramowania.

PEK_U04 Posiada praktyczne podstawy programowania obiektowego i przetwarzania danych i w języku Python.

PEK_U05 Potrafi poruszać się w tekstowej konsoli Linux

PEK_U06 Posiada praktyczne podstawy programowania w języku Bash

PEK_U07 Potrafi opisać architekturę chmury obliczeniowej.

PEK_U08 Potrafi pisać proste pliki Makefile, rozwiązujące zależności pomiędzy plikami źródłowymi.

PEK_U09 Posiada praktyczne umiejętności rozwiązywania podstawowych problemów programistycznych z wykorzystaniem technologii Bash i Python

Z zakresu kompetencji społecznych:

PEK_K01 – ma świadomość wpływu zarządzania wersjami oprogramowania (Software Configuration Management) na integralność, odtwarzalność i jakości tworzonego oprogramowania.

TREŚCI PROGRAMOWE

Forma zajęć - wykład		Liczba godz.
Wy1	Omówienie metodyki Scrum jako przykładu zwinnej metodyki prowadzenia projektów informatycznych.	2
Wy2	Podstawowe informacje o architekturze, sposobie działania oraz funkcjonalności systemu Linux	2
Wy3	Podstawy pracy w tekstowej konsoli Linux oraz podstawy skryptowania w języku Shell	2
Wy4	Podstawy programowania w języku Bash: instrukcje kontrolne, funkcje, operacje na plikach, tekstach, strumieniach, łańcuchach znaków i liczbach.	2
Wy5	Podstawy programowania w języku Bash: działania arytmetyczne, zaawansowane operacje na zmiennych, wyrażenia regularne.	2

Wy6	Wprowadzenie do systemów kontroli wersji. Charakterystyka Subversion jako jednego z najpopularniejszych systemów kontroli wersji.	2
Wy7	GIT jako przykład rozproszonego systemu kontroli wersji. Cechy oraz możliwości GIT'a, jako nowoczesnego systemu kontroli wersji. GIT Workflows - różne warianty pracy z narzędziem GIT.	2
Wy8	Wprowadzenie do GNU Make: cele, zastosowania, składnia, zasada działania. Przykłady recept.	2
Wy9	Podstawy programowania w języku Python: podstawowe typy danych, podstawowe operacje na liczbach i napisach, instrukcje sterujące, funkcje, realizacja obiektowości (klasy, metody, dziedziczenie).	2
Wy10	Podstawy programowania w języku Python: struktury danych (listy, słowniki, krotki), operacje na strukturach danych, obsługa modułów, wyjątki, podstawowe operacje na plikach.	2
Wy11	Omówienie koncepcji oraz architektury chmury obliczeniowej, jej przykładowych implementacji.	2
Wy12	Wprowadzenie do systemów Continuous Integration.	2
Wy13	Charakterystyka systemów CI na przykładzie narzędzia Jenkins/Hudson. Alternatywne systemy CI.	2
Wy14	Umieszczenie tematyki poszczególnych wykładów w kontekście pracy dużego projektu programistycznego. Omówienie zależności pomiędzy obszarami prezentowanymi podczas zajęć w ramach ich wspólnej implementacji.	2
Wy15	Wykład podsumowujący, zaliczenie końcowe	2
	Suma godzin	30

Forma zajęć - laboratoria		Liczba godzin
L1	Szkolenie stanowiskowe BHP. Sprawy organizacyjne. Wprowadzenie – przygotowanie stanowisk do ćwiczeń semestralnych.	2
L2	Praktyczne wykorzystanie metodyk zwinnych na przykładach z codziennego życia projektu programistycznego. Workshop.	2
L3	Poruszanie się po strukturze katalogów systemu Linux, kontrola uprawnień, zarządzanie procesami.	2
L4	Używanie komend powłoki systemowej Linux. Operacje na plikach.	2
L5	Podstawowe konstrukcje w Bashu: warunki, pętle, funkcje. Operacje na plikach, tekstach, łańcuchach znaków, liczbach.	2
L6	Bash: działania arytmetyczne, zaawansowane operacje na zmiennych, wyrażenia regularne.	2
L7	Praktyczne użycie systemu kontroli wersji SVN. Uruchomienie serwera SVN lokalnie, wstępna konfiguracja, połączenie z serwerem SVN, użycie podstawowych komend systemu.	2
L8	Praca z systemem GIT. Użycie głównych elementów takich jak: sub-moduły, gałęzie, remote'y, wskaźniki.	2
L9	GNU Make: tworzenie własnych prostych plików Makefile dla gotowych plików źródłowych.	2
L10	Poznanie interpretera języka Python. Wykorzystanie podstawowych instrukcji języka Python (print, import, help, dir)	2
L11	Operacje na typach danych w języku Python. Tworzenie struktur danych (lista, słownik, krotka), operacje na strukturach danych, kontrola przepływu danych (if, for, while)	2
L12	Tworzenie własnych funkcji w języku Python. Operacje na plikach, obsługa	2

	wyjątków.	
L13	Praktyczne wykorzystanie wiedzy pozyskanej na zajęciach z tematów Bash i Python do rozwiązywania zadań programistycznych	2
L14	Automatyzacja pracy przy pomocy Jenkins CI	2
L15	Laboratorium podsumowujące – wykorzystanie zdobytych wcześniej umiejętności przy rozwiązywaniu złożonych problemów	2
	Suma godzin	30

STOSOWANE NARZĘDZIA DYDAKTYCZNE	
N1. Wykład z użyciem środków multimedialnych N2. Prezentacja syntetyczna problematyki ćwiczeń (ok. 10 min - przez prowadzącego) N3. Ćwiczenia praktyczne przy użyciu komputerów z dyskusją rozwiązań zadań N4. Konsultacje N5. Praca własna – przygotowanie do ćwiczeń N6. Praca własna – samodzielne studia, przygotowanie do końcowego sprawdzianu	

OCENA OSIĄGNIĘCIA PRZEDMIOTOWYCH EFEKTÓW KSZTAŁCENIA

Oceny (F – formująca (w trakcie semestru), P – podsumowująca (na koniec semestru))	Numer efektu kształcenia	Sposób oceny osiągnięcia efektu kształcenia
F1	PEK_W01- PEK_W06, PEK_K01	Obecność i aktywność na wykładach (1 nieobecność dozwolona), ocena z pisemnego kolokwium końcowego.
F2	PEK_U01-PEK_U06, PEK_K01	Obecność i aktywność na ćwiczeniach (2 nieobecności dozwolone), łączna ocena z testów cząstkowych.
$P = F1 * 0,6 + F2 * 0,4$		

LITERATURA PODSTAWOWA I UZUPEŁNIAJĄCA
<p><u>LITERATURA PODSTAWOWA:</u></p> <p>[1] Matyas S., Glover A., <i>Continuous Integration: Improving Software Quality and Reducing Risk</i>, Addison-Wesley, 2008</p> <p>[2] Lutz M., Ascher D., <i>Python. Wprowadzenie</i>, O'Reilly, 2010</p> <p>[3] Myers G. J., Sandler C., Badgett T., Thomas T.M., <i>Sztuka testowania oprogramowania</i>, Helion 2005</p> <p>[4] Dawson M., <i>Python Programming for the Absolute Beginner</i>, Cengage Learning, 2010</p> <p>[5] Ziadé T., <i>Expert Python Programming: Learn Best Practices to Designing, Coding, and Distributing Your Python Software</i>, Packt Publishing, 2008</p> <p><u>LITERATURA UZUPEŁNIAJĄCA:</u></p> <p>[1] Hetland M. L., <i>Python Algorithms: Mastering Basic Algorithms in the Python Language</i>, Apress, 2010</p> <p>[2] Martin R. C., <i>Czysty kod. Podręcznik dobrego programisty</i>, Helion, 2010</p> <p>Pozycje desygnowane przez wykładowcę na zakończenie każdego wykładu.</p>
OPIEKUN PRZEDMIOTU (IMIĘ, NAZWISKO, ADRES E-MAIL)
Dr inż. Iwona Poźniak- Koszałka, iwona.pozniak-koszalka@pwr.edu.pl